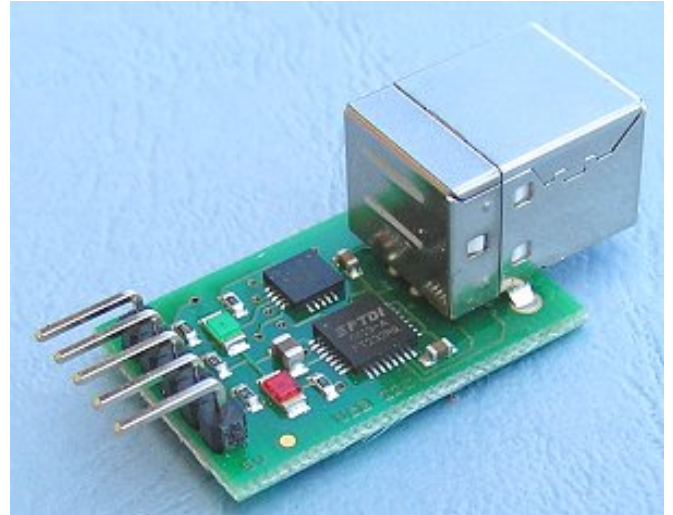


## 1.- INTRODUCCION

La empresa Inglesa Devantech Ltd. pone a disposición de nuestros clientes y usuarios el módulo de comunicaciones USB-I2C que permite hacer de puente entre el bus USB de cualquier PC y un bus I2C estándar. Se alimenta desde el propio bus USB y es capaz de alimentar dispositivos I2C externos con +5Vcc y hasta 70mA. El módulo USB-I2C a todos los efectos se comporta como un dispositivo I2C en el modo Master. Por un lado se conecta al puerto USB y, por el otro, dispone de un conector con las señales propias del bus I2C donde se conecta cualquier dispositivo I2C esclavo estándar o cualquiera de los desarrollados por la propia firma Devantech.Ltd. como son:



- SRF02: Medidor ultrasónico de distancias de bajo coste
- SRF08: Medidor ultrasónico de distancias
- CMPS03: Compás o brújula electrónica
- TPA81: Sensor térmico
- SP03: Sintetizador de voz
- MD03: Driver controlador de un motor de CC
- MD22: Driver controlador de 2 motores de hasta 50V/5A
- MD23: Driver controlador de 2 motores de 12VDC / 3 A. Con codificadores.
- SD21: Controlador de hasta 21 servo motores.

Se dispone así de un sistema que permite gobernar dispositivos I2C y obtener información de los mismos desde un PC con puerto USB y dotado de un simple software de comunicaciones. Es ideal para aplicaciones orientadas al control en general y a la robótica en particular.

## 2.- LOS DRIVERS

El corazón del módulo USB-I2C es el chip FTDI FT232BM USB (<http://www.ftdichip.com> ). La documentación que proporciona FTDI es muy completa y en su web podemos descargar las versiones más actualizadas de su software.

Antes de conectar el módulo USB-I2C es necesario instalar los drivers “FTDI’s Virtual COM Port (VCP)”, gracias a los cuales se consigue un nuevo puerto COM virtual que se añade a los ya existentes en el PC. De esta manera los programas genéricos de comunicación o los de aplicación del propio usuario acceden al dispositivo USB de la misma manera que si accediera a un puerto COM estándar de Windows mediante el empleo de las funciones VCOMM API o bien usando las librerías apropiadas.

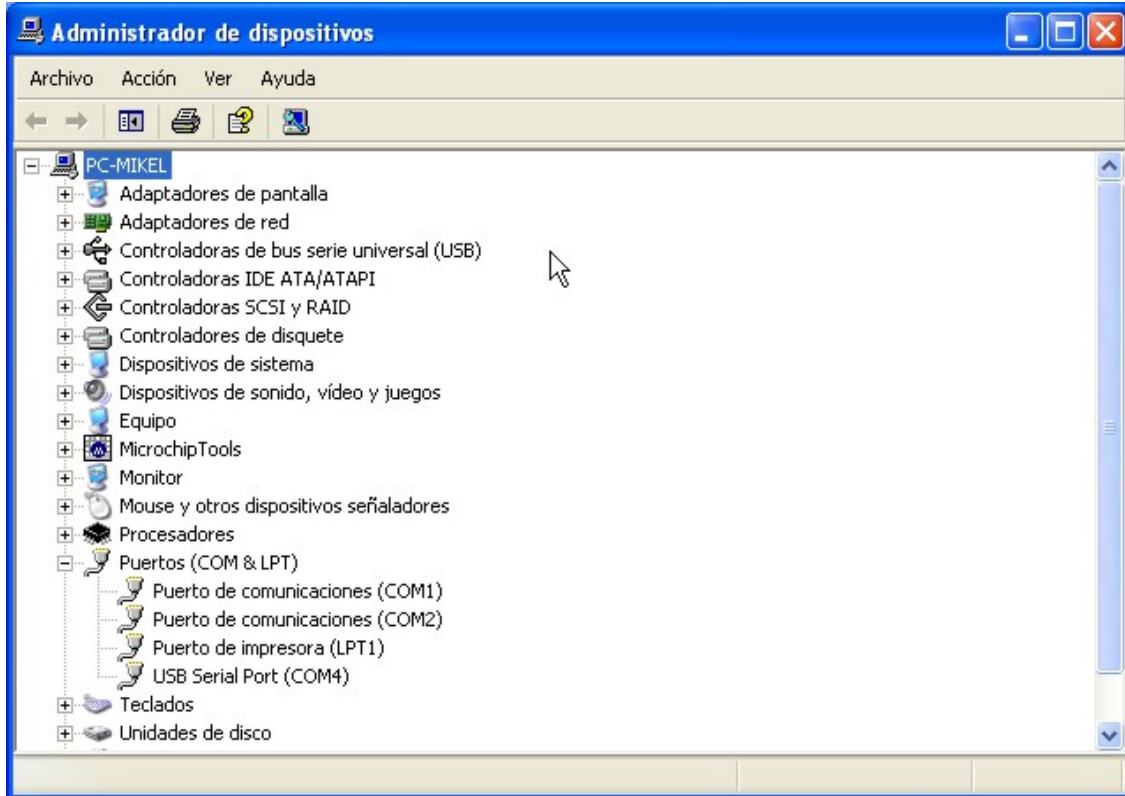
Estos Drivers están disponibles para Windows, Apple y Linux en la dirección <http://www.ftdichip.com/FTDriver.htm>.

## 3.- EL CANAL COM

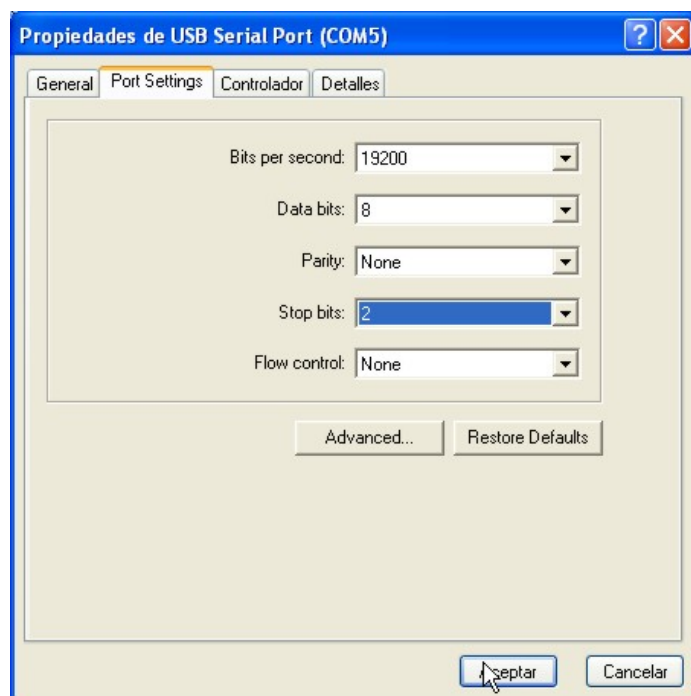
Una vez instalados los drivers, podemos conectar el USB-I2C en cualquier puerto USB disponible y será automáticamente reconocido. Ahora debemos localizar qué canal serie COM le ha sido asignado. Esto puede variar de un sistema a otro ya que depende de los puertos COM que estén actualmente instalados en el PC.

La forma de averiguarlo es desde “**Mi PC**” -> “**Panel de Control**” -> “**Sistema**” -> “**Administrador de dispositivos**”, que nos visualiza un esquema con todos los dispositivos y recursos actuales del PC. Abriendo el desplegable correspondiente a “**Puertos “COM & LPT”**” nos presenta en qué COM virtual fue asignado el

dispositivo USB. Por ejemplo, tal y como se muestra en la figura, el dispositivo USB se ha asociado al puerto serie COM4.



Por último queda por determinar las propiedades del nuevo puerto COM. Basta con seleccionarlo en la lista de la figura anterior, pulsar el botón derecho del ratón y seleccionar **“Propiedades”**. En **“Port Settings”** elegimos 19200 baudios de velocidad, 8 bits de datos, sin paridad y dos bit de stop. Ver la figura siguiente.



### 3.1 El software Hiperterminal

Cualquier software de comunicaciones se puede emplear para la transferencia de datos entre el PC y el USB-I2C. Por ejemplo podemos emplear el software Hiperterminal que se incluye en Windows. Habrá que seleccionar el puerto COM virtual al que está conectado el módulo y ajustar las propiedades de comunicación, que debe ser las mismas que se establecieron anteriormente: 19200 baudios, 8 bits de datos, sin paridad y 2 bit de stop.

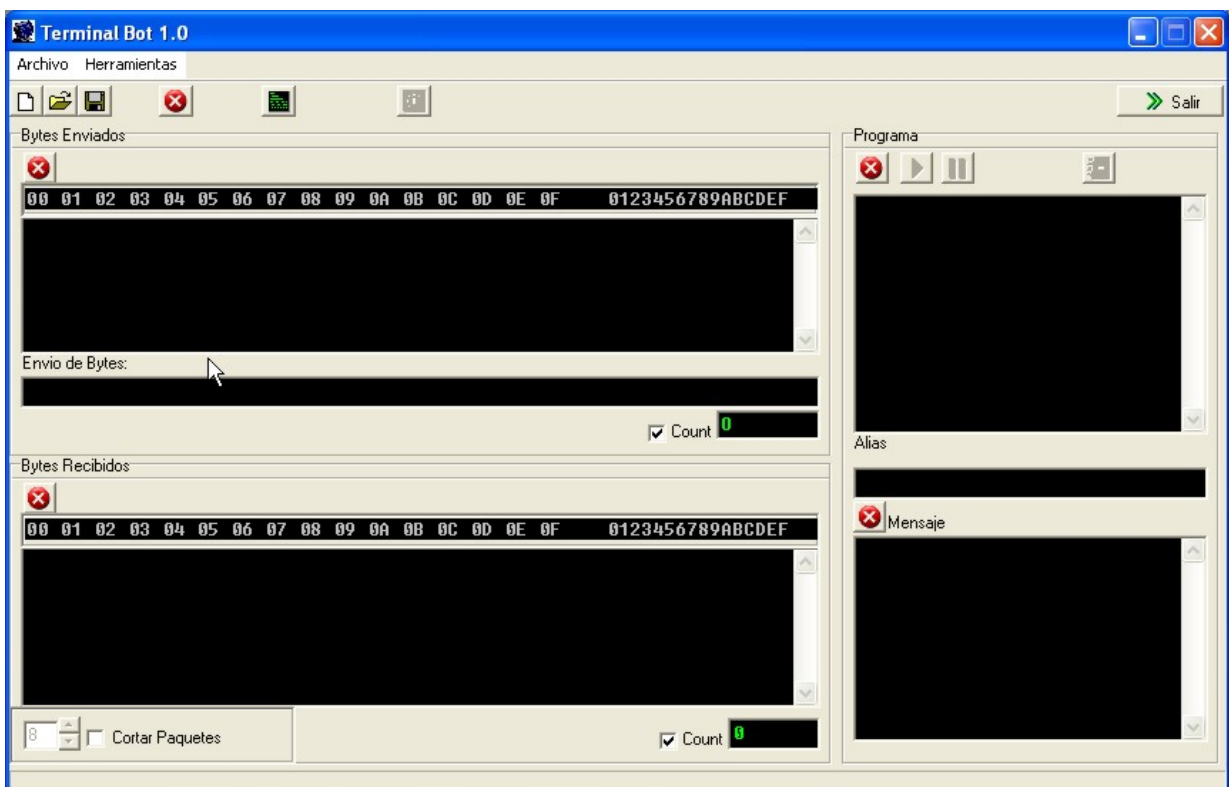
El único problema es que el Hiperterminal es un software de comunicaciones tipo Terminal y sólo reconoce caracteres ASCII. Sin embargo en muchas ocasiones el dispositivo que queremos gobernar sólo interpreta y transfiere códigos hexadecimales por lo que en caso de utilizar el Hiperterminal será necesario traducir esos códigos hex. en sus equivalentes en decimal. Por ejemplo si se quiere mandar el código 0x2A hay que traducirlo a decimal (42) y mandarlo manteniendo pulsada la tecla Alt al tiempo que se teclea 42 desde el teclado numérico.

### 3.2 El software Termbot

Este es un software experimental en fase beta desarrollado por Ingeniería de Microsistemas Programados S.L. y empleado en sus laboratorios para uso interno. Puede servir como programa Terminal de comunicaciones con la diferencia de que lo que se manda y recibe son siempre códigos o bytes hexadecimales. Para el caso que nos ocupa puede ser muy válido. Para otros casos el usuario deberá crear su propio software diseñado a medida.

El Termbot ocupa un único fichero ejecutable y es muy fácil de usar. Cada vez que se ejecuta presenta un primer cuadro de diálogo preguntando por el puerto COM que se va a utilizar. Si seguimos con nuestro ejemplo, debemos seleccionar COM4. Un segundo cuadro de diálogo permite establecer las propiedades de la comunicación. Para el módulo USB-I2C estas deben ser 19200,N,8,2. Esto es, 19200 baudios, sin paridad, 8 bits de datos y 2 bit de stop.

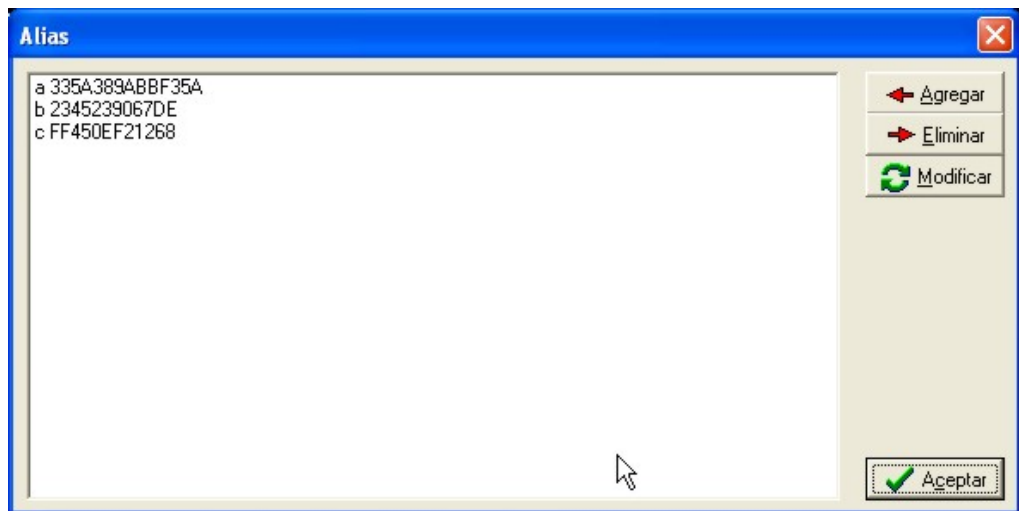
Aparece una ventana de trabajo como la que se muestra en la figura siguiente:



Tenemos una línea para el Envío de Bytes. Aquí se teclean, uno tras otro, los códigos hexadecimales que se desean transmitir. Todos ellos son enviados cuando se pulsa Enter. Los bytes enviados se visualizan en la ventana superior. También disponemos de un contador que nos indica el total de bytes enviados.

En la ventana inferior irán apareciendo, en hexadecimal, los bytes recibidos según van llegando. Un contador refleja el total de estos.

A la derecha disponemos de una línea para escribir el Alias. Un Alias no es más que una etiqueta a la que se le asocia una serie de códigos hexadecimales. Para transmitirlos basta simplemente con teclear ese Alias. Esto resulta especialmente cómodo cuando estamos controlando un dispositivo al que hay que transmitirle una secuencia de bytes contiguos.



Cuando se pulsa Ctrl. + L se accede a una ventana como la que se muestra en la figura y que permite editar, añadir o eliminar diferentes Alias. En el ejemplo se aprecian los siguientes Alias: a, b y c. Cada uno de ellos lleva asociado una secuencia de varios bytes. Para transmitir cualquiera de esas secuencias basta con teclear el Alias correspondiente. El USB-I2C tiene un buffer interno de 60 bytes que es el máximo número de bytes que se pueden transmitir en un único paquete.

#### **4.- CONEXIONES**

El módulo USB-I2C hace de puente entre el PC y el bus I2C. Por un lado se conecta al puerto USB del PC mediante un conector USB tipo B. Por otra parte dispone de 5 señales que se emplean para conectarlo con los diferentes dispositivos I2C que se desean gobernar. Ver la figura adjunta.



##### **4.1 GND**

Conexión de tierra de alimentación

##### **4.2 Entrada In 1**

Esta patilla se puede emplear como una línea de entrada digital con resistencia Pull-Up de 47K. Si la aplicación no la necesita, esta línea se puede dejar sin conexión.

##### **4.3 Línea SCL I/O 2**

La señal SCL se corresponde con la señal de reloj del bus I2C. La genera el propio módulo USB-I2C que siempre actúa como master. Si este módulo no se conecta a un bus I2C, esta señal se comporta como una línea de E/S de propósito general. Mediante los comandos SETPINS y GETPINS se puede actuar sobre el periférico que estuviera conectado. También se puede emplear como entrada analógica con una resolución de 10 bits que se puede leer mediante el comando GETAD.

#### 4.4 Línea SDA I/O3

Es la señal de datos del bus I2C. Por ella se transfiere información hacia o desde los dispositivos I2C a controlar y el módulo USB-I2C y de aquí al PC. Si este módulo no se conecta a un bus I2C, esta señal se comporta como una línea de E/S de propósito general. Mediante los comandos SETPINS y GETPINS se puede actuar sobre el periférico que estuviera conectado. También se puede emplear como entrada analógica con una resolución de 10 bits que se puede leer mediante el comando GETAD.

#### 4.5 +5VCC

Salida de tensión de alimentación de +5VCC. Puede alimentar dispositivos I2C externos con un consumo máximo de hasta 70mA. Si se necesita una mayor intensidad se recomienda no emplear esta línea y utilizar una fuente de alimentación externa. En cualquier caso por esta patilla no debe aplicarse nunca +5VCC.

### 5.- COMANDOS

El módulo USB-I2C acepta una serie de comandos procedentes del PC al que está conectado a través de un puerto USB. Por su parte, se entiende que dicho PC está dotado de un software de comunicaciones capaz de realizar transferencias de bytes. Este software puede ser de carácter genérico como el "Hiperterminal", el "Termbot", o cualquier otro similar, o bien puede ser un software específico de aplicación diseñado por el usuario.

Estos comandos se agrupan en tres categorías principales:

**I2CD\_CMD (0x53):** Permite leer o escribir dispositivos I2C que no dispongan registros internos a los que direccionar. Un ejemplo puede ser el dispositivo PCF8574. Se trata de un circuito de ampliación de líneas de E/S de propósito general que no dispone de ningún registro interno. Basta realizar un ciclo de lectura/escritura sobre el mismo para acceder y actuar sobre las E/S.

**I2C\_CMD (0x55):** Permite leer o escribir dispositivos I2C que dispongan registros internos como puede ser una memoria EEPROM o una RAM o cualquier otro al que se necesite direccionar sus diferentes registros internos: registros de comandos, registros de configuración, registros de estado, registros de datos, etc..

**USB-I2C\_CMD (0x5A):** Permite la ejecución de una serie de sub comandos propios del módulo USB-I2C y orientados fundamentalmente al control y empleo de otros módulos de la firma Devantech

#### 5.1 Escribiendo un dispositivo I2C sin registros internos

En este tipo de dispositivos (p.e. PCF8574) sólo es necesario enviarles su propia dirección I2C de esclavo y el byte de datos.

	Comando primario del USB-ICD	Dirección del dispositivo I2C + bit de R/W	Byte de datos
<b>Tipo</b>	I2CD_CMD	Dirección + bit R/W	Dato
<b>Ejemplo</b>	0x53	0x40	0x00
<b>Descripción</b>	Comando principal	Dirección I2C del PCF8574 en modo escritura	Poner a "0" todas las líneas de E/S

El módulo USB-I2C recibe desde el PC una única secuencia de 3 bytes. En el ejemplo se trata de poner a "0" las líneas de E/S de un dispositivo PCF8574 cuya dirección I2C se supone es 0x40. La secuencia de tres bytes necesaria es: 0x53, 0x40, 0x00. El primer byte (0x53) se corresponde con el comando primario del módulo USB-I2C, el 2º byte (0x40) corresponde la dirección I2C del esclavo al que se desea acceder en el modo escritura (p.e. PCF8574), el 3er. Byte (0x00) es el dato a escribir (p.e. poner a 0 las líneas de E/S).

Una vez que el USB-I2C recibe la secuencia de tres bytes, procede a la ejecución del comando y devuelve al PC un valor de retorno: 0x00 si se produce un fallo en la ejecución del comando o cualquier otro valor si la ejecución es correcta. El software de control en el PC debe esperar el retorno de este byte antes de proceder a enviar la siguiente secuencia con el siguiente comando.

## 5.2 Leyendo un dispositivo I2C sin registros internos

La secuencia de bytes es muy similar a la explicada anteriormente. En este caso sólo es necesario indicar, en el modo lectura, la dirección I2C del esclavo al que se accede.

I2CD_CMD	Dirección + bit R/W
0x53	0x41

En el ejemplo se desea leer el estado de las líneas de E/S del dispositivos PCF8574 cuya dirección I2C en el modo de lectura es 0x41. La secuencia a enviar al módulo USB-I2C está formada en esta ocasión por dos bytes (0x53,0x41). El primer byte (0x53) corresponde con el comando del módulo USB-I2C propiamente dicho y el 2º (0x41) se corresponde con la dirección del esclavo PCF8574 en el modo de lectura.

Una vez que el comando se recibe y se ejecuta, el módulo USB-I2C devuelve al PC un byte que representa el valor procedente de las líneas de E/S del PCF8574 recién leído. El software de control en el PC debe esperar la llegada de este byte (unos 500mS) antes de proceder a enviar la siguiente secuencia con el siguiente comando.

## 5.3 Escribiendo un dispositivo I2C con registros internos

Es una de las operaciones clásicas a realizar con cualquier dispositivo I2C. Tras el comando primario I2C\_CMD (0x55) se manda la dirección del dispositivo en modo escritura, la dirección del registro interno sobre el que se quiere escribir, el número de bytes a escribir a partir de esa dirección interna y los bytes a escribir propiamente dichos. En total, el máximo número de bytes a transferir no debe ser superior a 60 para no sobrepasar el buffer interno del CM02. Sirva como muestra el ejemplo siguiente:

	Comando primario del CM02	Dirección del dispositivo I2C + bit de R/W	Dirección del registro internos	Nº de bytes de datos a escribir	Bytes de datos
<b>Tipo</b>	I2C_CMD	Dirección + bit R/W	Posición interna inicial	Contador de bytes	Datos
<b>Ejemplo</b>	0x55	0xE0	0x00	0x01	0x51
<b>Descripción</b>	Comando del USB-I2C	Dirección del SRF08 en modo escritura	Registro de comandos del SRF08	Se va a escribir 1 byte	Comando para iniciar una nueva medida en cm

Este ejemplo muestra una secuencia de 5 bytes recibida por el USB-I2C y transmitida por el PC mediante un software de comunicaciones. Los 5 bytes (o los que sean) deben ser transmitidos y recibidos por el USB-I2C de forma contigua uno tras otro. En caso contrario, transcurrido un tiempo, se reinicia el sistema y se ignora todo el mensaje.

En este ejemplo, cuando se reciben todos los bytes el módulo USB-I2C ejecuta la operación I2C. Esta consiste en seleccionar el dispositivo SRF08 (dirección 0xE0) en modo escritura, y escribir en su registro de comandos (registro interno 0x00) un único byte (0x51) que inicia una nueva medida de distancias en cm. Si la escritura sobre el dispositivo SRF08 se realiza con éxito, el módulo USB-I2C retransmite un valor distinto de 0x00. Si se produce algún error en la escritura sobre el SRF08, se retransmite el valor 0x00. El PC debe esperar a que llegue este byte de control (aprox. 500mS) antes de proceder con la siguiente transferencia.

Este ejemplo se supone dirigido a controlar los motores conectados al dispositivo I2C MD22 (Driver para controlar dos motores). La secuencia transmitida por el PC y el software de comunicaciones y, recibida por el USB-I2C, debe ser algo parecido a esto:

I2C_CMD	Dirección del MD22 en modo escritura	Dirección del registro interno de modo	Nº de bytes a escribir	Modo 1 de trabajo del MD22	Motor 1 parado	Motor 2 parado	Aceleración
0x55	0xB0	0x00	0x04	0x01	0x00	0x00	0x02

Si el módulo USB-I2C responde con un valor distinto de 0 es que la escritura sobre el MD22 se realizó correctamente, si responde con 0x00 hubo un error.

#### 5.4 Leyendo un dispositivo I2C con registros internos

La secuencia es similar excepto que el último bit correspondiente a la dirección del dispositivo I2C debe estar a nivel "1" para definir que se va a realizar una lectura del mismo. Para leer un dispositivo SRF08 cuya dirección es la 0xE0, la dirección real será 0xE1. El máximo nº de bytes que se deseen leer no debe ser superior a 60 para no sobrepasar el buffer interno del módulo USB-I2C.

En el siguiente ejemplo se muestra la secuencia necesaria para leer el dispositivo CMPS03 (compás o brújula electrónica):

I2C_CMD	Dirección del CMPS03 en modo lectura	Dirección del primer registro interno a leer	Nº de bytes a leer
0x55	0xC1	0x02	0x02

El módulo USB-I2C realiza un ciclo de lectura sobre el bus I2C y re-envía al PC dos bytes procedentes del dispositivo CMPS03 (posición respecto al norte magnético). El PC debe esperar a recibir estos 2 bytes antes de proceder a la siguiente transferencia.

#### 5.5 Comandos propios del USB-I2C

Al margen de ejecutar ciclos de lectura/escritura sobre cualquier dispositivo I2C como se ha visto anteriormente, el módulo USB-I2C tiene su propio juego de sub-comandos. Gracias a estos es posible cambiar direcciones I2C, averiguar el estado de las baterías, etc.. También permiten ejecutar de forma rápida y cómoda ciertas operaciones relacionadas con algunos dispositivos I2C del mismo fabricante como son leer varios medidores ultrasónicos SRF08, leer la posición indicada por el compás CMPS03, actuar sobre el controlador de motores MD22 y mucho mas. Todo ello facilita enormemente tareas orientadas al control de sistemas móviles o robots.

El formato de estos comandos es el siguiente:

CM02_CMD	Sub comandos del USB-I2C	Dato 1	Dato 2
0x5A	Ver la lista	Específico de cada sub comando	Específico de cada sub comando

Los comandos propios del CM02 consisten siempre en una secuencia de 4 bytes. Empieza con el código 0x5A, que es el comando primario, continúa con el código de uno de los sub comandos disponibles y finaliza con 2 bytes de datos que pueden o no ser válidos según el comando en cuestión, pero que hay que mandarlos.

La siguiente tabla resume los sub comandos disponibles:

Hex.	Comando	Bytes devueltos	Descripción
0x01	REVISION	1	Devuelve la versión del firmware interno
0x02	DIRECCION	1	Cambia la dirección I2C de los dispositivos SRF08
0x03	NO USADO	1	Empleado por compatibilidad con el módulo CM02. Devuelve 0x00
0x04	SCAN1	6	Establece velocidad de motores y devuelve estado de la batería, compás y sónar (SRF08).
0x05	SCAN2	9	Lo mismo pero se suponen 2 dispositivos SRF08
0x06	SCAN3	12	Igual pero con 3 dispositivos SRF08
0x07	SCAN4	15	4 dispositivos SRF08
0x08	SCAN6	21	6 dispositivos SRF08
0x09	SCAN8	27	8 dispositivos SRF08
0x0A	SCAN12	39	12 dispositivos SRF08
0x0B	SCAN16	51	16 dispositivos SRF08
0x10	SETPINS	1	Activa/Desactiva las líneas de I/O 2 - I/O 3
0x11	GETPINS	1	Lee el estado de las líneas In 1, I/O2 e I/O 3
0x12	GETAD	4	Lee el valor analógico presente en I/O2 – I/O 2

### 5.5.1 Revisión

Este sub comando se emplea para conocer la versión del firmware interno del USB-I2C. Devuelve un único byte indicando el número de versión. Los dos bytes de datos, Dato 1 y Dato 2, son irrelevantes pero hay que incluirlos en el comando. La secuencia a enviar es: **0x5A 0x01 0xDato1 0xDato2**

### 5.5.2 Dirección

Este permite cambiar de una forma muy sencilla la dirección I2C de los medidores ultrasónicos SRF08. La nueva dirección se debe indicar en el byte Dato 1. El valor de Dato 2 es irrelevante pero hay que incluirlo. Para ejecutar correctamente este comando es necesario que en ese instante haya sólo un SRF08 conectado al bus I2C, el que se desee cambiar de dirección. Supongamos que se desea cambiar la dirección de un SRF08 (por defecto 0xE0) por la dirección 0xE6. La secuencia de bytes a enviar al CM02 es: **0x5A 0x02 0xE6 0xDato2**

### 5.5.2 No usado

Este comando no tiene ninguna utilidad. Se mantiene por compatibilidad con el módulo CM02 que es similar al USB-IC pero trabajando de forma inalámbrica junto con el transceptor RF04.

### 5.5.3 SCAN

Este comando está pensado expresamente para el control de una forma fácil y rápida de un Robot móvil que emplea distintos módulos y Drivers de la firma Devantech Ltd. Se asume que dicho robot consta de un Driver MD22 para el control de los motores, un compás o brújula electrónica CMPS03 y uno o varios medidores ultrasónicos SRF08: SCAN1 asume que se dispone de un SRF08, SCAN2 de dos, SCAN3 de tres y así sucesivamente.

Los dos bytes de datos, Dato 1 y Dato 2, que acompañan al comando, se emplean para expresar los valores de velocidad para los dos motores controlados por el Driver MD22. Una vez enviado el comando al USB-I2C, este ejecuta de forma automática los ciclos de lectura/escritura necesarios sobre los diferentes módulos que se presumen conectados al bus I2C. De esta forma el módulo USB-I2C devuelve una trama de varios bytes con los resultados obtenidos. La trama empieza con un byte que informa de la tensión de la batería, los siguientes dos bytes indican la posición respecto al norte magnético obtenida gracias al compás CMPS03 y a continuación tres bytes por cada SRF08 disponible. El primero de esos tres bytes expresa el valor del sensor de luz y los siguientes dos bytes la distancia medida respecto al objeto u obstáculo.

A modo de ejemplo se muestra a continuación la trama que se recibe cuando el USB-I2C recibe y ejecuta el comando **0x55 0x05 0x80 0x80** (comando SCAN2 aplicando el valor de velocidad 0x80 a los dos motores controlados por el MD22).

Tensión batería	Posición CMPS03 byte alto	Posición CMPS03 byte bajo	Sensor de luz del 1er SRF08 dir. 0xE0	Byte alto de la distancia del 1er SRF08 en dir. 0xE0	Byte bajo de la distancia del 1er SRF08 en dir. 0xE0	Sensor de luz del 2º SRF08 dir. 0xE2	Byte alto de la distancia del 2º SRF08 en dir. 0xE2	Byte bajo de la distancia del 2º SRF08 en dir. 0xE2

Los valores correspondientes a los SFR08 se supone proceden de estos dispositivos cuyas direcciones deben ser contiguas y comenzar desde 0xE0 (0xE0, 0xE2, 0xE4, etc...). Los valores que estos devuelven expresan microsegundos. Dividir entre 58 para convertir a cm o entre 148 para convertir a pulgadas.

Para ejecutar este comando se supone que las direcciones I2C de los diferentes SRF08 que se van a emplear deben estar debidamente ajustadas. Igualmente el driver MD22 debe haberse configurado previamente respecto al modo de trabajo y aceleración.

El comando SCAN pone en marcha un temporizador interno de 500mS. Si durante este tiempo no se reciben nuevos comandos SCAN, el USB-I2C ejecuta un comando para detener automáticamente los motores.

De esta forma se evita que el robot o la plataforma móvil que se está controlando pueda quedarse sin control si, por ejemplo, falla la comunicación entre el PC y el módulo USB-IC.

## **6.- LOS LED**

El módulo USB-I2C dispone de dos leds de estado. El rojo indica la alimentación del sistema y el verde parpadea cada vez que se realizan transferencias de datos entre el USB-I2C y el PC. El led rojo se puede activar/desactivar mediante el comando SETPINS.

## **7.- LINEAS DE E/S**

El módulo USB-I2C, en lugar de actuar de puente entre el PC y los dispositivos I2C, se puede emplear como un controlador de E/S de propósito general para el control de periféricos.

**In 1:** Esta línea siempre es entrada y tiene asociada una resistencia pull-up de 47K.

**I/O 2 – I/O 3:** Estas dos líneas pueden ser de entrada o salida. Las salidas se activan/desactivan mediante el comando SETPINS. Se pueden poner a “1” mediante una resistencia pull-Up interna de 4,7K o a nivel “0” con una corriente máxima de 24mA. El comando GETPINS devuelve el estado de estas líneas. Para usar una de estas líneas como entrada es necesario ponerlas previamente a nivel “1”

Ambos comandos SETPINS y GETPINS devuelven el estado de las líneas de E/S, pero únicamente el comando GETPINS las puede modificar. El formato de los bits escritos mediante SETPINS y leídos con GETPINS es el siguiente:

7	6	5	4	3	2	1	0
X	X	X	X	I/O 3	I/O 2	In 1	Led Rojo

El siguiente ejemplo permite desactivar el led rojo y configurar las líneas de E/S como entradas:

Comando USB-I2C	Comando SETPINS	Dato 1	Dato 2
0x5A	0x10	0x0E	0x00 (no válido)

### 7.1 Entradas analógicas

El módulo USB-I2C puede también convertir valores analógicos presentes en las líneas I/O2 e I/O3. Previamente estas líneas deben configurarse como entradas. Recordar que estas líneas están conectadas a +5VCC a través de sendas resistencias Pull-Up.

El siguiente ejemplo muestra la secuencia de bytes necesarias para realizar una conversión:

Comando USB-I2C	Comando GETAD	Dato 1	Dato 2
0x5A	0x12	0x00 (no válido)	0x00 (no válido)

El módulo USB-I2C responde con los siguientes bytes:

Byte 1	Byte 2	Byte 3	Byte 4
Byte alto de I/O 2	Byte bajo de I/O 2	Byte alto de I/O 3	Byte bajo de I/O 3

Como resultado de la conversión se ofrece un valor de 10 bits comprendido entre 0x0000 y 0x03FF.

***¡ IMPORTANTE ! No puede combinarse el modo I2C con el modo de E/S. Los comandos dedicados a la gestión de las líneas de E/S no deben emplearse si hubiera algún dispositivo I2C conectado, ya que se emplean las mismas señales.***

## 8.- SOFTWARE DE TEST

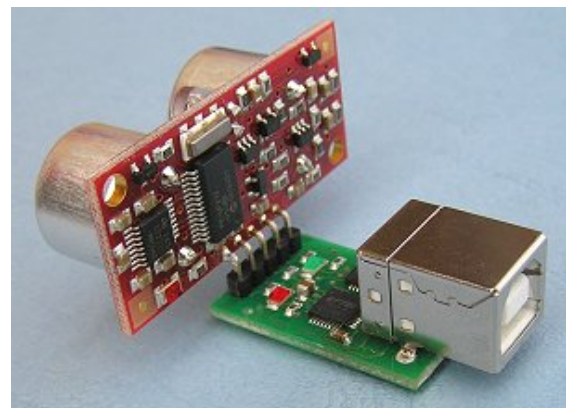
Para facilitar la comprobación del módulo USB-I2C, Devantech Ltd. pone a disposición de todos los usuarios dos programas de test que se pueden descargar desde su sitio web y también desde nuestra página [www.microcontroladores.com](http://www.microcontroladores.com). Ambos programas se suministran en versión ejecutable para PC y también en código fuente, que puede servir como modelo para el desarrollo propio de aplicaciones.

Para su ejecución se supone que los drivers del USB-I2C, que convierten el puerto USB al que está conectado en un puerto COM virtual, están debidamente instalados. Se conecta el USB-I2C al puerto USB y se ejecuta la aplicación.

### 8.1 Controlando un SRF08

Este ejemplo pretende demostrar el funcionamiento del módulo USB-I2C controlando el medidor de distancias ultrasónico SRF08 que actúa como un dispositivo I2C esclavo. Ver la figura.

Las 5 líneas del USB-I2C se conectan directamente con el SRF08. Desde el mismo puerto USB del PC se toma la alimentación tanto para el propio módulo USB-I2C como para el sensor SRF08.



El programa ejecutable se puede descargar desde la dirección web [http://www.robot-electronics.co.uk/files/USB\\_I2C\\_SRF08.exe](http://www.robot-electronics.co.uk/files/USB_I2C_SRF08.exe) y el código fuente desde la dirección [http://www.robot-electronics.co.uk/files/usb\\_i2c\\_srf08.zip](http://www.robot-electronics.co.uk/files/usb_i2c_srf08.zip). La ejecución de este programa de ejemplo presenta una ventana como la mostrada en la figura.



En primer lugar se debe seleccionar el número de canal serie virtual, según lo explicado en el apartado 3. El programa busca automáticamente un dispositivo SRF08 y visualiza su dirección I2C. También se muestra la versión del firmware interno del módulo USB-I2C.

Este programa de aplicación está constantemente tomando muestras de la distancia medida por el sensor ultrasónico SRF08. El resultado se visualiza en pulgadas, cm y  $\mu$ S. También se visualiza la medida de luz ambiente que incide sobre la célula LDR que dispone el SRF08.

## 8.2 Programa de test para el USB-I2C

Este programa es una simple demostración del funcionamiento de las líneas de E/S del propio módulo USB-I2C. El código ejecutable se puede descargar desde [http://www.robot-electronics.co.uk/files/usb\\_i2c\\_io.exe](http://www.robot-electronics.co.uk/files/usb_i2c_io.exe) y el código fuente desde [http://www.robot-electronics.co.uk/files/usb\\_i2c\\_io.zip](http://www.robot-electronics.co.uk/files/usb_i2c_io.zip). Su ejecución presenta una ventana como la de la figura.

Seleccionado el canal serie adjudicado como puerto COM virtual, el programa visualiza el estado actual de las líneas de entrada y el valor analógico de las mismas. También permite modificar el valor de las líneas I/O 2 e I/O 3.

